

## ***Development of Matching the Card Educational Game using iOS-based Shuffle Random Algorithm***

Fajar Haryadi <sup>1</sup>, Pitrasacha Adytia <sup>2</sup>, dan Wahyuni <sup>3</sup>

Teknik Informatika, STMIK Widya Cipta Dharma  
Jl. M. Yamin No. 25, Samarinda, Kalimantan Timur, 75123  
E-mail: [fajarharyadi.31@gmail.com](mailto:fajarharyadi.31@gmail.com) <sup>1</sup>, [pitra@wicida.ac.id](mailto:pitra@wicida.ac.id) <sup>2</sup>, [wahyuni@wicida.ac.id](mailto:wahyuni@wicida.ac.id) <sup>3</sup>

### **ABSTRACT**

*Games are one of the implementations of computer science that has developed very rapidly. Games are also a form of educational application, meaning they can be used as a learning medium where the process can be carried out with the concept of learning while playing for children. Matching the Card is an educational game that tests memory by matching pairs of hidden cards. This game uses a random shuffle algorithm to randomize the position of the cards, creating new challenges in each session. The purpose of this development is to provide entertainment media that trains the user's memory and concentration in a fun way. With a simple interface and random shuffling mechanism, this game is designed to be dynamic and interesting.*

*In this study, the method used to design software is the Multimedia Development Life Cycle (MDLC) development method, which involves stages starting from conceptualization, design, material collecting, assembly, testing and distribution, as well as the use of tools such as the Unified Modeling Language (UML). The algorithm used is shuffle randomization where players cannot memorize the position of the playing image and make the game not boring and are able to apply a learning system while playing and become a good learning method among children aged 7-9 years because in this age range cognitive abilities, especially memory and concentration, are developing rapidly.*

*The results of this study, based on beta testing that has been carried out on 10 student respondents, show that this educational game matching the card produces an average value of 84 which is included in the EXCELLENT category with a grade scale of A. This indicates that the Educational Game Matching the Card Using the iOS-Based Shuffle Random Algorithm can be used effectively. And in the black box test showed good results without any indication of error.*

**Keywords:** *game, ios, education, algorithm, shuffle, matching, card.*

---

## ***Pengembangan Game Edukasi Matching the Card menggunakan Algoritma Shuffle Random berbasis iOS***

### **ABSTRAK**

*Game adalah salah satu implementasi bidang ilmu komputer yang perkembangannya sudah sangat pesat. Game juga merupakan bentuk aplikasi yang edukatif, artinya bisa dijadikan sebagai media pembelajaran dimana prosesnya bisa dilakukan dengan konsep belajar sambil bermain untuk anak-anak. Matching the Card adalah permainan edukatif yang menguji daya ingat dengan mencocokkan pasangan kartu tersembunyi. Permainan ini menggunakan algoritma shuffle random untuk mengacak posisi kartu, sehingga menciptakan tantangan baru di setiap sesi. Tujuan dari pengembangan ini adalah menyediakan media hiburan yang melatih memori dan konsentrasi pengguna secara menyenangkan. Dengan antarmuka sederhana dan mekanisme pengacakan acak, permainan ini dirancang agar dinamis dan menarik.*

*Dalam penelitian ini metode yang digunakan untuk merancang perangkat lunak adalah metode pengembangan Multimedia Development Life Cycle (MDLC), yang melibatkan tahap-tahap mulai dari pengonsepan (concept), perancangan (design), pengumpulan materi (material collecting), pembuatan (assembly), pengujian (testing) dan pendistribusian (distribution), serta penggunaan alat bantu seperti Unified Modeling Language (UML). Algoritma yang digunakan adalah pengacakan shuffle dimana pemain tidak dapat menghafal posisi gambar bermain dan membuat permainan tidak membosankan serta mampu menerapkan sistem belajar sambil bermain dan menjadi metode pembelajaran yang baik dikalangan anak-anak dengan rentang usia 7-9 tahun karena pada rentang usia ini kemampuan kognitif, terutama daya ingat dan konsentrasi, sedang berkembang pesat.*

*Hasil dari penelitian ini, berdasarkan pengujian beta testing yang telah dilakukan kepada 10 responden siswa/i, menunjukkan bahwa game edukasi matching the card ini menghasilkan nilai rata-rata 84 yang tergolong ke dalam kategori EXCELLENT dengan grade scale A. Hal tersebut mengindikasikan bahwa Game Edukasi Matching the Card Menggunakan Algoritma Shuffle Random Berbasis iOS dapat digunakan secara efektif. Serta dalam pengujian black box menunjukkan hasil yang baik tanpa adanya indikasi eror.*



**Kata Kunci:** *game, ios, edukasi, algoritma, shuffle, matching, card.*

## 1. Pendahuluan

*Matching the Card* merupakan jenis *game* edukasi yakni *game* yang dirancang untuk melatih kemampuan kognitif, memori, dan konsentrasi pemain terutama anak-anak usia dini hingga sekolah dasar dengan cara mencocokkan pasangan-pasangan gambar yang sama. Pemain hanya dapat membuka dua gambar secara bersamaan. Jika kedua gambar tersebut sama, kartu akan tetap terbuka, jika berbeda, kartu akan kembali tertutup. *Game* ini sering digambarkan sebagai proses *flip* (membalik kartu), dan biasanya dilakukan dalam keterbatasan waktu. Meskipun sederhana, konsep ini terbukti efektif dalam meningkatkan daya ingat, kemampuan berpikir kritis, serta keterampilan pemecahan masalah.

Perkembangan teknologi saat ini terus melaju dengan pesat dan telah menjadi kebutuhan bagi sebagian besar masyarakat, terutama di bidang pembelajaran. Salah satu pemanfaatan teknologi dalam proses pembelajaran adalah pengembangan *game* edukasi. *Game* edukasi dapat diartikan sebagai media permainan interaktif yang memuat materi pembelajaran untuk membantu mendidik dan mengarahkan siswa dalam proses pembelajaran yang menyenangkan (Winarni, Naimah, & Widiyawati, 2020).

Banyak pengguna *Platform* berbasis *iOS* (*iPhone Operating System*) mengeluhkan keterbatasan akses terhadap fitur-fitur tertentu yang hanya dapat dibuka melalui pembelian dalam aplikasi (*in-app purchase*). Hal ini sering kali menimbulkan kesenjangan antara ekspektasi pengguna terhadap *game* gratis dan kenyataan bahwa beberapa fitur utama memerlukan biaya tambahan. Selain itu, proses publikasi di *App Store* memiliki regulasi yang ketat yang wajib dipatuhi. Ini berpotensi menjadi hambatan bagi peneliti atau pengembang independen karena biaya dan aturan yang cukup ketat. Dalam pengembangan aplikasi juga menghadapi tantangan dalam menyediakan *game* gratis yang berkualitas tanpa mengorbankan keberlanjutan finansial dari aplikasi tersebut. Oleh karena itu, penting untuk mengeksplorasi solusi inovatif yang dapat menjembatani kebutuhan pengguna akan aksesibilitas serta kebutuhan pengembang akan profitabilitas.

Sebagai solusi penelitian menggunakan sehingga peneliti dapat meminimalkan kebutuhan perangkat fisik dengan memanfaatkan simulator bawaan *Xcode* dan penginstalan melalui *mode developer* pada perangkat *iPhone* untuk pengujian awal. Meskipun tidak menggantikan sepenuhnya, simulator dan *mode developer* dapat membantu mendeteksi bug dasar atau kesalahan logika. Untuk mendukung mekanika permainan, algoritma *Shuffle Random* digunakan sebagai cara mengacak urutan indeks dari sebuah *array*. Proses ini diibaratkan seperti mengocok dek kartu, sehingga susunan setiap elemen teracak secara tidak terprediksi. Sebagai contoh, pada *array*  $A = [1, 2, 3, 4, 5]$ , algoritma *Shuffle Random* dapat menghasilkan  $A1 = [5, 1, 3, 2, 4]$  atau susunan lain yang berbeda. Dalam bahasa pemrograman, fungsi *Shuffle*

*Random* tidak hanya dapat mengacak deretan angka, tetapi juga *array* berisi string ataupun gabungan keduanya.

*Game* edukasi ini diharapkan dapat memancing minat belajar siswa sekaligus memberikan pengalaman bermain yang menyenangkan. Dengan perasaan senang, siswa dapat lebih mudah memahami materi yang disajikan. Jenis *game* ini sesungguhnya lebih mengacu pada tujuan dan isi pembelajaran yang terkandung dalamnya, bukan sekadar *genre*. *Game* edukasi juga berpotensi menumbuhkan kembali motivasi belajar anak yang mengalami penurunan.

## 2. Batasan Masalah

Agar permasalahan dalam penelitian ini tidak terlalu meluas dan sulit untuk dipahami, maka perlu untuk menentukan batasan-batasan masalah, berikut beberapa batasan masalah yaitu:

Adapun batasan-batasan masalah penelitian ini sebagai berikut :

1. Aplikasi edukasi *game* ini hanya difokuskan pengembangannya pada pengguna satu perangkat saja yaitu perangkat berbasis *iOS*.
2. Aplikasi edukasi *game* ini menggunakan alat pemrograman *Xcode*.
3. Aplikasi edukasi *game* ini hanya dapat dijalankan menggunakan simulator *Xcode* dan penginstalan melalui *mode developer*.
4. Aplikasi edukasi *game* ini ditujukan untuk anak-anak Sekolah Dasar kelas 2 dengan rentang usia 7-9 tahun.
5. Edukasi pada *game* ini menggunakan objek buah.

Metode pengembangan yang digunakan dalam penelitian ini adalah *Multimedia Development Life Cycle (MDLC)*

## 3. Dasar Teori

Dalam penelitian ini diperlukan sebuah atau suatu konsep dalam merumuskan definisi-definisi yang menunjang penelitian ini, baik secara teori dasar maupun teori pada umumnya.:

### 3.1 Permainan Edukasi

“Sebuah *game* atau permainan merupakan aktivitas yang dijalankan berdasarkan aturan tertentu, di mana selalu ada pihak yang menang dan kalah. Biasanya, permainan ini disajikan dalam konteks yang santai dengan tujuan utama untuk memberikan hiburan dan menyegarkan pikiran.”. Macam - macam *game*, antara lain:

1. Aksi

*Game* ini macam *game* yang paling populer. *Game* jenis ini membutuhkan kemampuan refleks pemain.

2. Aksi petualangan

*Game* ini memadukan *game play* aksi petualangan. Contohnya pemain diajak untuk menyelusuri gua bawah tanah sambil mengalahkan musuh atau menyeberangi sungai.

3. Simulasi,

Konstruksi dan Manajemen Pemain dalam *game* ini diberi keleluasaan untuk membangun dan suatu proyek tertentu dengan bahan baku yang terbatas.

#### 4. Olahraga

*Game* ini membawa olahraga ke dalam sebuah komputer atau konsol. Biasanya *game play* dibuat semirip mungkin dengan kondisi olahraga yang sebenarnya.

#### 5. Puzzle

*Game puzzle* menyajikan teka-teki, menyamakan warna bola, perhitungan matematika, menyusun balok, atau mengenal huruf dan gambar.

#### 6. Permainan Kata

*Word game* sering dirancang untuk menguji kemampuan dengan bahasa atau untuk mengeksplorasi sifat - sifatnya. *Word game* umumnya digunakan sebagai sumber hiburan, tetapi telah dibuktikan untuk melayani suatu tujuan pendidikan juga (Firdausi, 2020).

### 3.2 Matching the Card / Match-Up Game

*Match-Up* adalah permainan yang dirancang untuk mengasah ingatan pemain melalui pencocokan gambar atau jawaban secara akurat. Semakin banyak pasangan gambar dan jawaban yang tersedia, tantangan dan keasyikan dalam permainan ini pun semakin meningkat. Permainan ini menawarkan keunggulan berupa kesempatan bagi siswa untuk belajar suatu konsep atau topik sambil bermain, sehingga menciptakan lingkungan pembelajaran yang menyenangkan dan merangsang semangat. Selain itu, aktivitas ini juga membantu meningkatkan konsentrasi siswa dalam menjawab pertanyaan.

Permainan ini mengasah memori pemain dengan menebak pasangan gambar yang identik. Setiap giliran, pemain hanya diperbolehkan membuka dua gambar sekaligus, dan jika kedua gambar tersebut cocok, mereka akan tetap terbuka. Namun, apabila tidak sesuai, kedua gambar akan ditutup kembali. Seringkali permainan ini disajikan dalam bentuk *flip* kartu dan dilengkapi dengan elemen waktu untuk menambah tantangan.

Objek-objek gambar permainan *match-up* biasanya disusun dengan susunan matriks seperti pada Gambar 1. Semakin banyak jumlah pasangan gambar, semakin sukar permainan ini untuk di selesaikan. (Andrea, 2015).

?	?	?	?	?	1
?	?	?	?	?	?
2	2	?	?	?	?
?	?	?	1	?	?

Gambar 1. Permainan *Match-up* mencocokkan angka  
 Figure 1. *Match-up* game matching numbers

### 3.3 Algoritma *Shuffle Random*

*Shuffle random* merupakan metode pengacakan urutan indeks dalam sebuah *array* atau *record*, serupa dengan cara kita mengocok sebuah dek kartu sehingga setiap elemennya tersusun secara acak. Misalnya, jika terdapat *array* A berukuran 5 x 1 dengan  $A = [1, 2, 3, 4, 5]$ , maka proses

*shuffle random* dapat mengubah urutannya menjadi  $A_1 = [5, 1, 3, 2, 4]$  atau bentuk susunan lain yang acak.

Fungsi *shuffle random* dalam bahasa pemrograman tidak terbatas hanya pada pengacakan angka; fungsi ini juga dapat diterapkan pada *array* yang berisi *string*, maupun *array* yang merupakan campuran antara *string* dan angka (Heri et al., 2021).

### 3.4 iOS

*iOS* adalah sistem operasi seluler yang dikembangkan oleh *Apple*, perusahaan teknologi ternama. Nama *iOS* merupakan kependekan dari *iPhone Operating System*, yang dirancang khusus untuk menjalankan perangkat *iPhone*. Dapat dilihat pada Gambar 2 logo *iOS* seperti namanya, semula *iOS* dikembangkan untuk mendukung pengoperasian produk perangkat telepon genggam buatan *Apple*, *iPhone*. *iOS* yang sebelumnya disebut dengan *iPhone OS*, awalnya dibuat khusus untuk tiga perangkat *Apple* seperti *iPhone*, *iPad*, dan *iPodTouch*.

Pada tahun 2007, sistem *iOS* pertama kali diluncurkan, yang pada awalnya menghadirkan revolusi dalam industri telepon seluler. Saat itu, ponsel-ponsel populer masih mengandalkan tombol fisik, sedangkan *Apple* memperkenalkan *iPhone* pertamanya dengan antarmuka layar penuh.

Pada generasi *iPhone* pertama, *iOS* berhasil mengintegrasikan berbagai fungsi dalam satu perangkat. Sistem operasi ini memungkinkan pengguna untuk mengirim *email*, melakukan penjelajahan *internet*, dan memutar musik dalam satu perangkat. Namun, versi awal *iOS* belum menyediakan fitur toko aplikasi *App Store*.

Pada tahun 2008, *Apple* meluncurkan *App Store*, sebuah toko aplikasi khusus untuk *iOS*. Pada saat yang bersamaan, *Apple* juga memperkenalkan *Software Development Kit (SDK)* guna mendukung para pengembang dalam menciptakan aplikasi untuk perangkat *iOS*.

Setiap tahunnya, *Apple* merilis versi pembaruan *iOS* yang dapat langsung diakses melalui menu Pengaturan di *iPhone*. Hingga kini, *iOS* terus mengalami pengembangan dan pembaruan secara berkala. Proses update pun menjadi lebih praktis, hanya memerlukan koneksi *internet* yang stabil dan daya baterai yang mencukupi.



Gambar 2. Logo *Apple iOS*  
 Figure 2. Logo *Apple iOS*

## 4. Alat Bantu Pengembangan Sistem

### 4.1 Xcode



*Xcode* adalah suatu alat pemrograman besutan *Apple* untuk membuat aplikasi yang dijalankan dalam lingkungan *Apple* (*iOS* dan *MacOS*). Alat pemrograman seperti *Xcode* ini biasa disebut juga dengan *Integrated Development Environment (IDE)*, seperti *Visual Studio* dan *Xcode* mendukung beberapa bahasa pemrograman seperti *C*, *C++*, *Objective C* dan yang terakhir mendukung bahasa *Swift* yang tersedia pada *Xcode*. Seperti pada umumnya *IDE*, maka *Xcode* menyediakan semua kebutuhan pemrograman seperti *editor*, *compiler*, *debugger* dan *run-time simulator*.

*Xcode* menyediakan fitur manajemen pengembangan aplikasi yang meliputi *repository* dan *version control*. Dengan integrasi ke *git*, *Xcode* memudahkan pelaksanaan *software configuration management* dalam lingkungan kerja tim ((Frasha & Masitoh, 2023).

Selain itu, *Xcode* juga memiliki *tools* yang memudahkan kamu saat melakukan *coding*, beberapa di antaranya seperti:

1. *Editor*

Semua aktivitas *coding*, dari awal hingga uji testing terakhir berada pada fitur *Editor* ini. Maka, cocok jika dikatakan jika *Editor* adalah *tools* paling penting dari *Xcode*.

2. *Simulator*

Dengan *tools Simulator* pada *Xcode* ini, kamu dapat melakukan simulasi pada aplikasi yang telah kamu buat. Terdapat tombol *play* and *stop* dan *item drop down* yang membantu kamu untuk menguji coba fungsi dari aplikasi yang dibuat. Jenis simulatornya pun bisa kamu pilih tergantung pada jenis aplikasi apa yang kamu kembangkan.

3. *Status Bar*

Saat mengembangkan sebuah aplikasi, kamu pasti ingin melihat sudah sejauh mana progres yang telah kamu buat. Nah, pada Bilah Status (*Status Bar*) kamu bisa melihat sejauh mana progres dan letak kesalahan yang kamu buat, sehingga memudahkan kamu untuk memantau selama proses pengerjaan.

4. *View*

*Xcode* dilengkapi dengan berbagai tombol *View* yang bisa dipakai untuk membandingkan dua *file Swift* secara berdampingan, atau menyembunyikan panel sisi kiri dan kanan agar kamu bisa fokus pada bagian tengah editor *Swift* saja.

5. *Debug*

*Debugging* adalah fungsi penting dari semua perangkat lunak *IDE*, termasuk *Xcode*. *Debugging* digunakan untuk memperbaiki bug pada aplikasi. Logo aplikasi *Xcode* dapat dilihat pada Gambar 3.



Gambar 3. Logo Xcode

Figure 3. Logo Xcode

4.2 Swift

*Swift* adalah bahasa pemrograman yang dikembangkan oleh *Apple* untuk membangun aplikasi berbasis *iOS*, *macOS*, *watchOS* dan *tvOS*. *Swift* dikembangkan menggunakan bahasa pemrograman *C* dan *Objective C*. *Swift* merupakan bahasa pemrograman yang ideal bagi pemula karena mudah dipahami dan memiliki sintaks ekspresif, sehingga proses pembelajaran terasa lebih menarik. Selain itu, *Swift* dilengkapi dengan fitur *playgrounds*, yang memungkinkan programmer untuk bereksperimen dan melihat hasil secara langsung tanpa harus melalui proses build atau menjalankan aplikasi secara keseluruhan.

Dapat dilihat pada Gambar 4 logo *Swift* mencerminkan perpaduan antara konsep pemrograman modern dan budaya rekayasa *Apple*. *Apple* telah mengoptimalkan kinerja *compiler* serta mempermudah proses pengembangan baik untuk aplikasi sederhana seperti "*Hello World*" maupun aplikasi yang kompleks tanpa mengorbankan performa. Hal ini menjadikan *Swift* sebagai investasi berharga bagi para *developer* dan masa depan *Apple*.

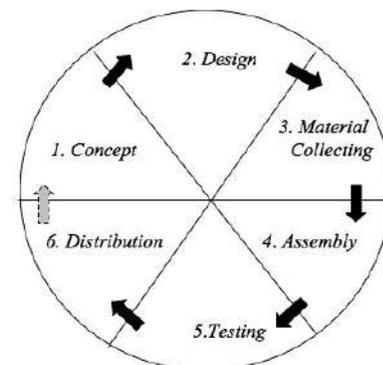


Gambar 4. Logo Swift  
Figure 4. Logo Swift

5. Metode Pengujian Sistem

5.1 Metode Multimedia Development Life Cycle (MDLC)

Metode pendekatan sistem yang digunakan pada penelitian ini adalah metode *Multimedia Development Life Cycle (MDLC)*. Pada metode ini terdapat enam tahapan yaitu pengonsepan (*concept*), perancangan (*design*), pengumpulan materi (*material collecting*), pembuatan (*assembly*), pengujian (*testing*) dan pendistribusian (*distribution*) (Mustika et al., 2018).



Gambar 5. Tahapan Multimedia Development Life Cycle  
Figure 5. Multimedia Development Life Cycle Stages

Berikut merupakan 6 (enam) tahapan kegiatan *Multimedia Development Life Cycle (MDLC)* yang dilakukan, yaitu :

#### 1. *Concept*

Tahap konsep merupakan langkah awal dalam pengembangan, di mana tujuan program dan identifikasi target pengguna ditetapkan. Pada tahap ini, juga dilakukan penentuan jenis aplikasi—apakah berupa presentasi, interaktif, dan sebagainya serta mendefinisikan tujuan utama aplikasi, misalnya untuk hiburan, pelatihan, atau pembelajaran.

#### 2. *Design*

*Design* (perancangan) adalah tahap membuat spesifikasi mengenai arsitektur program, gaya, tampilan maupun kebutuhan material atau bahan yang akan digunakan untuk membangun program yang akan digunakan.

#### 3. *Material Collecting*

*Material collecting* adalah tahap pengumpulan bahan yang diperlukan sesuai dengan kebutuhan proyek. Proses ini dapat berlangsung bersamaan (paralel) dengan tahap *assembly*. Namun, dalam beberapa situasi, *material collecting* dan *assembly* dilakukan secara berurutan (linear) dan tidak secara paralel.

#### 4. *Assembly*

*Assembly* atau tahap pembuatan merupakan proses penciptaan semua elemen multimedia yang telah dirancang. Pada tahap ini, aplikasi dikembangkan berdasarkan desain yang telah dibuat, dan hasilnya adalah sebuah program atau aplikasi yang dapat berfungsi dengan baik.

#### 5. *Testing*

Setelah tahap *assembly* selesai, tahap pengujian dilakukan dengan menjalankan program untuk memastikan tidak ada kesalahan yang muncul.

#### 6. *Distribution*

Pada fase distribusi, aplikasi disimpan ke dalam media penyimpanan. Jika kapasitas media penyimpanan terbatas sehingga tidak dapat menampung aplikasi secara utuh, maka dilakukan proses kompresi untuk mengurangi ukuran file aplikasi tersebut.

### 5.2 Unified Model Language (UML)

*Unified Modeling Language (UML)* adalah sebuah alat bantu dalam pemodelan yang mendukung perancangan sistem berorientasi objek. Dengan menggunakan UML, proses spesifikasi, visualisasi, dan dokumentasi sistem dapat dilakukan secara efektif selama fase pengembangan (Rohmah, 2002).

*Unified Modeling Language (UML)* Merupakan bahasa visual yang dirancang untuk memodelkan dan menyampaikan informasi mengenai suatu sistem melalui penggunaan diagram dan teks pendukung (Desyanti & Wetri Febrina, 2020).

### 5.3 Pengertian Use Case Diagram

*Use Case Diagram* adalah pemodelan untuk kelakuan (*behavior*) *Use Case Diagram* menggambarkan interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dikembangkan, sehingga memudahkan pemodelan dan komunikasi mengenai fungsionalitas sistem (Kristianingrum & Al-Fadillah, 2022).

### 5.4 Activity Diagram

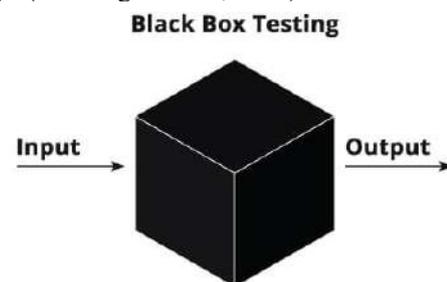
Penjelasan tentang simbol-simbol yang digunakan dalam diagram aktivitas terdapat pada Tabel 2. *Activity Diagram* adalah diagram state khusus di mana mayoritas *state* merepresentasikan aksi, dan transisi antar *state* dipicu oleh penyelesaian state sebelumnya dalam proses internal. Oleh karena itu, diagram ini tidak menguraikan perilaku internal sistem atau interaksi antar subsistem secara acak, melainkan memberikan gambaran umum mengenai proses dan alur aktivitas di tingkat atas. (Ananda Muhammad Tri Utama, 2022).

### 5.5 Sequence Diagram

*Sequence Diagram* merupakan representasi *visual* yang menggambarkan urutan aktivitas atau kolaborasi antar objek. Diagram ini berfungsi untuk menunjukkan bagaimana pesan atau interaksi dikomunikasikan antar setiap objek dalam sistem (Raharjana & Justitia, 2015).

### 5.6 Black Box Testing

*Black box testing*, yang juga dikenal sebagai *Behavioral Testing*, adalah metode pengujian perangkat lunak yang fokus pada hasil input dan output tanpa mempertimbangkan struktur kode internal. Pengujian ini biasanya dilakukan pada tahap akhir pembuatan perangkat lunak untuk memastikan bahwa sistem berjalan dengan baik sesuai fungsinya (Uminingsih et al., 2022).



Gambar 6. *Black Box Testing*  
Figure 6. *Black Box Testing*

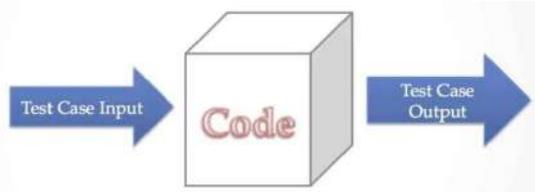
### 5.7 White Box Testing

*White box testing* merupakan sebuah metode pengujian yang dilakukan untuk menguji dan menganalisis kode program apabila terjadi kesalahan atau tidak. *White box testing* dilakukan dengan melihat kode program saja tanpa memperhatikan tampilan *user interface* dari halaman aplikasi (Bustomi & Adytia, n.d.).

Pada *white box testing* dikenal istilah *basis path*. *Basis path* merupakan salah satu teknik pengujian pada *white box testing*. *Basis path testing* lebih cocok digunakan dibandingkan dengan teknik lainnya karena *basis path testing* akan menghasilkan jumlah *test case* dengan cakupan *test* yang lebih menyeluruh dibandingkan teknik lainnya. Teknik ini memungkinkan perancang *test case* untuk menghasilkan pengukuran kompleksitas logika dari perancangan prosedural dan menggunakan pengukuran ini sebagai perkiraan untuk menguraikan jalur dasar eksekusi.

Teknik *basis path* terdiri dari *flowgraph notation* yang merupakan notasi sederhana yang menggambarkan alur kontrol program, *Cyclomatic Complexity (CC)* yang merupakan perhitungan untuk menentukan jumlah dari jalur pengujian, *independent path* yang merupakan penentuan jalur pengujian yang dilewati setidaknya sekali, dan *test*

case yang merupakan penentuan alur pengujian berdasarkan jalur *independent path* yang sudah ditentukan. Pada setiap *test case* kita perlu mendefinisikan output yang kita harapkan.



**Gambar 7. White Box Testing**  
Figure 7. White Box Testing

Tahapan *white box testing* pada penelitian ini ialah mengumpulkan kode program yang ingin diuji, membuat *flow graph*, perhitungan CC, setelah itu dilakukan penentuan jalur *independent path* dan terakhir melakukan pengujian. CC dapat dihitung dengan rumus berikut:

$$V(G) = E - N + 2$$

Keterangan :

$V(G)$  = Cyclomatic Complexity

$E$  = Edge

$2$   $N$  = Node

### 5.8 System Usability Scale (SUS)

*System Usability Scale (SUS)* adalah metode evaluasi yang mengukur tingkat kemudahan penggunaan suatu produk. Dikembangkan oleh John Brooke pada tahun 1986, teknik ini telah diterapkan untuk menilai berbagai produk dan layanan, mulai dari perangkat keras dan perangkat lunak, hingga perangkat seluler, situs web, serta aplikasi *mobile* (Sukma et al., 2023).

Pengujian antarmuka pengguna dan pengalaman pengguna (*UI/UX*) pada pengembangan game edukasi *iOS "Matching the Card"* yang menerapkan algoritma *shuffle random* dilakukan melalui kuesioner *System Usability Scale (SUS)*. Metode SUS ini merupakan instrumen yang digunakan untuk mengukur tingkat kepuasan pengguna dan telah disusun secara siap guna mendukung penelitian.

Dari kuesioner diatas *System Usability Scale (SUS)* menyediakan lima opsi jawaban, yaitu sangat tidak setuju, tidak setuju, ragu-ragu, setuju, dan sangat setuju, dengan skor yang diberikan berkisar dari 1 hingga 5, dapat dilihat pada Tabel 1.

**Tabel 1. Skor dari Jawaban**  
Table 1. Score of Answers

Answer	Score
Sangat Tidak Setuju (STS)	1
Tidak Setuju (TS)	2
Ragu-Ragu	3
Setuju (S)	4
Sangat Setuju (SS)	5

Setelah data dikumpulkan dari responden, data tersebut dihitung untuk memperoleh skor SUS. Dalam penggunaannya, terdapat beberapa aturan khusus yang

harus diikuti untuk menghitung skor *System Usability Scale (SUS)* dengan tepat. Berikut ini aturan-aturan saat perhitungan skor pada kuesionernya:

1. Untuk pertanyaan dengan nomor ganjil, nilai responden dikurangi 1.
2. Pada pertanyaan nomor genap, nilai akhir dihitung dengan mengurangkan skor responden dari angka 5.
3. Skor keseluruhan SUS diperoleh dengan menjumlahkan nilai setiap pertanyaan, lalu mengalikannya dengan 2,5.

Aturan perhitungan ini berlaku untuk setiap responden. Untuk mendapatkan skor keseluruhan, skor SUS masing-masing responden dijumlahkan dan kemudian dibagi dengan jumlah responden, menghasilkan nilai rata-rata. Rumus perhitungan skor *System Usability Scale (SUS)* ditunjukkan pada Persamaan 1.

$$\bar{x} = \frac{\sum x}{n}$$

Keterangan:

$\bar{x}$  = Skor rata-rata

$\sum x$  = Jumlah skor SUS

$n$  = Jumlah responden

## 6. Metode Penelitian

Penelitian ini merupakan penelitian pengembangan (*Research and Development/R&D*) yang bertujuan untuk merancang, membangun, dan menguji kelayakan serta efektivitas game edukasi *Matching the Card* berbasis *iOS*. Pendekatan ini digunakan untuk menghasilkan produk berupa aplikasi edukasi serta menguji kualitasnya baik dari aspek fungsionalitas maupun edukatif.

### 6.1 Waktu dan Lokasi Penelitian

Penelitian ini dilaksanakan di Sekolah Dasar Negeri 016 Tenggara, Kelurahan Loa Tebu, Kecamatan Tenggara, Kabupaten Kutai Kartanegara, Kalimantan Timur.

### 6.2 Teknik Pengumpulan Data

Dalam penelitian ini, ada beberapa cara/teknik pengumpulan data yang digunakan, antara lain :

1. Observasi

Observasi dilakukan dengan cara mengamati secara langsung agar mengetahui proses terkait penggunaan aplikasi game edukasi *Matching the Card* terhadap responden dengan siswa/i rentang usia 7-9 tahun di Sekolah Dasar Negeri 016 Tenggara.

2. Wawancara

Wawancara adalah salah satu teknik pengumpulan data dimana penulis dengan cara berhadapan langsung kepada responden dengan cara tanya jawab, agar dapat memperoleh data yang akurat untuk pengisian kuesioner.

3. Kuesioner

Kuesioner merupakan teknik pengumpulan data yang efisien sehingga menjadi tolak ukur dan tahu apa yang bisa diharapkan dari responden. Kuesioner berupa pertanyaan-

pertanyaan dapat diberikan kepada responden secara langsung.

#### 4. Studi Pusaka

Dilakukan kajian terhadap literatur yang relevan dengan topik sebagai referensi dan dasar pemahaman. Hal ini bertujuan untuk memperoleh landasan teori yang kokoh bagi sistem yang akan dikembangkan, sehingga penulisan tetap konsisten dengan teori-teori yang telah diakui kebenarannya. Kajian ini mencakup data tentang aplikasi monitoring serta kebutuhan informasi.

### 6.3 Metode Pengembangan Sistem

Pengembangan *Game* Edukasi *Matching the Card* berbasis iOS dengan Algoritma *Shuffle Random* ini mengadopsi metode *Multimedia Development Life Cycle (MDLC)* sebagai pendekatan pengembangannya.

Tahapan dari metode *Multimedia Development Life Cycle (MDLC)* pada penelitian ini yaitu sebagai berikut:

#### 6.3.1 Concept

Diperlukan konsep yang merinci tahapan pengembangan permainan, seperti identifikasi pengguna, spesifikasi umum aplikasi, dimensi, dan tipe aplikasi. Konsep ini nantinya akan dijadikan dasar dalam pengembangan aplikasi berbasis iOS. Dalam tahapan ini, terdapat tiga hal utama yang menjadi concept dalam membangun aplikasi ini, yaitu:

##### 1. Identifikasi Pengguna Aplikasi

Identifikasi target pengguna untuk aplikasi Pengembangan *Game* Edukasi *Matching The Card* Menggunakan Algoritma *Shuffle Random* Berbasis iOS. Target pengguna terdiri dari anak-anak dengan rentang usia 7-9 tahun.

##### 2. Analisis Kebutuhan

Kebutuhan yang diperlukan dalam pengembangan game edukasi ini mencakup aspek visual, mekanisme permainan, serta aset yang akan dirancang, seperti gambar kartu, efek suara, dan animasi interaktif untuk meningkatkan pengalaman bermain.

##### 3 Spesifikasi Umum

Berisi penjelasan tentang spesifikasi umum, minimum, serta rekomendasi perangkat dalam menjalankan aplikasi Pengembangan *Game* Edukasi *Matching The Card* Menggunakan Algoritma *Shuffle Random* Berbasis iOS.

##### 4 Analisis Perangkat Lunak

Spesifikasi dari perangkat lunak yang digunakan dalam pengembangan game edukasi ini, termasuk penggunaan *Xcode* sebagai *platform* pengembangan aplikasi berbasis iOS.

#### 6.3.2 Design

Tahapan ini mencakup perancangan detail mengenai struktur *game* yang akan dibuat. Desain dibuat secara rinci agar pada tahap berikutnya tidak memerlukan perubahan besar, melainkan menggunakan rancangan yang telah ditetapkan.

1. **Rancangan Antarmuka** *Game* ini memiliki beberapa rancangan antarmuka, seperti tampilan menu utama, tampilan permainan, tampilan skor, dan pengaturan permainan.
2. **Kontrol** Aplikasi ini dirancang untuk dapat dikontrol melalui sentuhan layar (*touchscreen*) pada kartu.

3. **Animasi dan Efek Visual** *Game* ini akan menggunakan animasi interaktif untuk memberikan pengalaman bermain yang menarik bagi pengguna.

#### 6.3.3 Material Collecting

Tahap *Material Collecting* merupakan proses pengumpulan dan pembuatan bahan yang diperlukan dalam pengembangan aplikasi. Material yang dikumpulkan terdiri dari berbagai elemen visual dan audio yang digunakan dalam game. Beberapa bahan dibuat sendiri menggunakan perangkat lunak pengolah grafis

Pengumpulan bahan untuk membuat aplikasi ini meliputi:

1. *Image* (Sprites dan Antarmuka Pengguna),
2. *Audio* (Suara dan Musik latar permainan),

Pengumpulan dan pembuatan material dilakukan secara paralel dengan tahap pengembangan game.

#### 6.3.4 Assembly

Tahapan *Assembly* merupakan tahapan implementasi atau integrasi dari seluruh komponen menjadi aplikasi fungsional. Proses ini mencakup:

1. Pengkodean dan Implementasi Algoritma *Shuffle Random*,
2. Integrasi *UI/UX* dengan Mekanisme Permainan,
3. Pengujian Awal pada Simulator iOS dan Perangkat Asli melalui *mode developer*.

#### 6.3.5 Testing

Tahapan pengujian dilakukan setelah tahap *assembly* selesai dan seluruh komponen telah dimasukkan ke dalam aplikasi visualisasi penelitian ini. Sebelum pengujian *alpha*, dilakukan pengujian *White Box Testing* untuk memastikan logika internal dan alur program berjalan sesuai dengan rancangan. Setelah itu, pengujian *alpha* dilakukan menggunakan metode *Black Box Testing* untuk memastikan fungsionalitas dasar permainan, berdasarkan evaluasi oleh penulis.

### 7. Pembahasan

Dalam sub bab ini penulis akan menjelaskan hasil dari metode pengembangan yang digunakan yaitu tahap pengembangan multimedia.

#### 7.1 Concept (Konsep)

Aplikasi game edukasi yang akan dibuat merupakan game *Matching the Card* dimana pengguna dapat meningkatkan daya ingat, logika, dan pemahaman pelajaran melalui permainan mencocokkan kartu bertema edukatif. Mengenalkan pengguna akan manfaat buah-buahan secara menyenangkan dan interaktif melalui permainan *Matching the Card*.

##### 7.1.1 Identifikasi Pengguna Aplikasi

Penggunaan game *Matching the Card* ini ditujukan untuk anak-anak Sekolah Dasar kelas 2 (dua) dengan rentang usia 7-9 tahun yang ingin merasakan keseruan edukasi dengan metode belajar berbasis visual dan permainan melalui smartphone berbasis iOS.

##### 7.1.2 Analisis Kebutuhan

Kebutuhan yang akan disiapkan pada penelitian ini diantaranya adalah: Perangkat berbasis *macOS* dan *iOS*, Aplikasi *Xcode* dan *Swift*, dan *Asset* yang terdiri dari gambar buah-buahan, latar belakang dan juga suara latar, untuk

menghidupkan dan mengisi latar *Matching the Card* dengan Penerapan *Game* Edukasi.

### 7.1.3 Analisis Perangkat Lunak dan Perangkat Keras

Spesifikasi dari perangkat lunak yang digunakan dalam pembuatan aplikasi *game* edukasi *matching the card* ini terlihat pada tabel 2.

**Tabel 2.** daftar perangkat lunak yang digunakan  
*Table 2. list of software used*

No.	Software Type	Software Name
1	Operating system	macOS Sequoia 15.4
2	Developer Tools	Xcode 16.3
3	Programming language	Swift 6.1

Spesifikasi dari perangkat keras yang digunakan dalam pembuatan aplikasi *game* edukasi *matching the card* ini terlihat pada tabel 3.

**Tabel 3.** daftar perangkat keras yang digunakan pada penelitian ini

*Table 3. list of hardware used in this study*

No.	Hardware Type	Hardware Name
1	Device	MacBook Air M2
2	Chip	Apple M2
3	Media storage	SSD 256 GM
4	Memory	RAM 8 GB

Spesifikasi dari perangkat *iOS* yang dibutuhkan dalam menjalankan aplikasi *game* edukasi *matching the card* ini terlihat pada tabel 4.

**Tabel 4.** daftar spesifikasi perangkat *iOS* yang dibutuhkan  
*Table 4. list of required iOS device specifications*

No.	Hardware Type	Hardware Name
1	Device	iPhone 14 Pro
2	Chip	A16 Bionic
3	CPU	6-core
4	GPU	5-core
5	Penyimpan media	256 GB
6	Memory	RAM 6 GB
7	Versi <i>iOS</i>	18.4.1
8	Kabel	USB Type C ke Lightning

## 7.2 Design

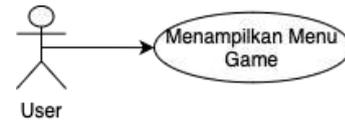
Pada Tahap desain ini merupakan langkah penting dalam menciptakan landasan yang kuat untuk implementasi selanjutnya. Sebagai hasilnya, kita mendapatkan pandangan yang lebih rinci tentang bagaimana proyek akan dijalankan dan bagaimana setiap elemen akan berinteraksi satu sama lain. Dengan demikian, tahap desain memberikan landasan yang kokoh sebelum melangkah ketahap implementasi yang sebenarnya, tahapannya diantaranya:

### 7.2.1 Konsep Permainan

Adapun perancangan UML dari *game* edukasi *matching the card* sebagai berikut:

#### 1. Use Case Diagram

*Use case diagram* menggambarkan tindakan yang akan diambil sistem untuk dibuat dan aktor yang akan berinteraksi dengan sistem. *Use case diagram* dapat dilihat pada gambar 8.

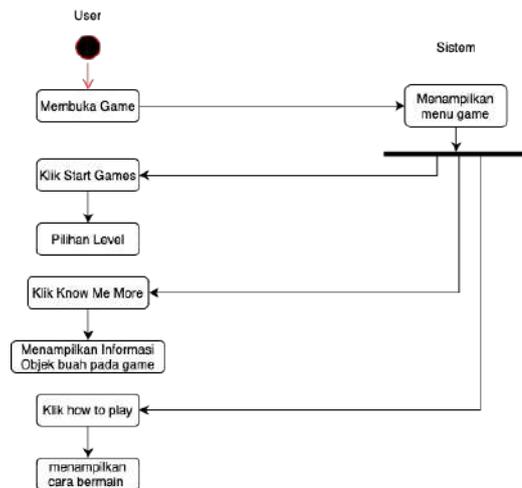


**Gambar 8.** Use case diagram  
*Figure 8. Use case diagram*

*Use case* akan dijelaskan bagaimana alur yang akan berjalan yang diusulkan yaitu, *user* merupakan *level* yang mempunyai semua hak akses untuk bermain. *User* memilih menu *game* untuk memulai permainan, kemudian *user* memilih *level* pilihan untuk menentukan tingkatan permainan yang akan di mainkan mulai dari *easy*, *medium* dan *hard*. Yang dimana masing – masing *level* tersebut memiliki perbedaan waktu yang menjadi tantangan bagi pemain, kemudian *user* memilih *game* dimulai untuk mulai memainkan permainan *game* edukasi.

#### 2. Activity Diagram

*Activity diagram* digambarkan pada berbagai aliran aktivitas dalam sistem yang dirancang, informasi tentang bagaimana setiap aliran dimulai, keputusan apa yang mungkin dibuat, dan bagaimana itu berakhir. Sebuah *use case* atau kasus penggunaan dapat digunakan untuk mewujudkan suatu aktivitas. *Use case* menggambarkan bagaimana *user* menggunakan sistem untuk melakukan aktivitas, sementara aktivitas menggambarkan proses yang dilakukan.

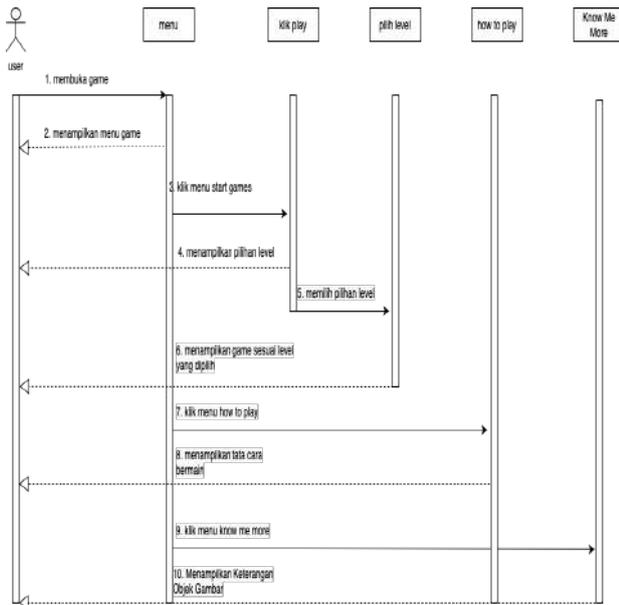


**Gambar 9.** Activity diagram  
*Figure 9. Activity diagram*

*Activity diagram* pada gambar 9 menjelaskan hubungan *user* dengan sistem. *User* membuka *game* kemudian sistem menampilkan menu *game* yang di dalamnya terdapat tampilan klik *Start Games* untuk menampilkan pilihan *level*, dan *Know Me More* untuk menampilkan Informasi objek buah yang digunakan pada permainan dan klik *How To Play* untuk panduan permainan.

#### 3. Sequence Diagram

*Sequence diagram* pada gambar 10 menjelaskan alur kerja objek dengan pesan yang dikirim dan digunakan untuk menjelaskan perilaku objek dalam *use case*. Untuk membangun *sequence diagram*, seseorang harus mengetahui objek yang terlibat dalam *use case* serta kelas yang diterapkan pada objek tersebut.



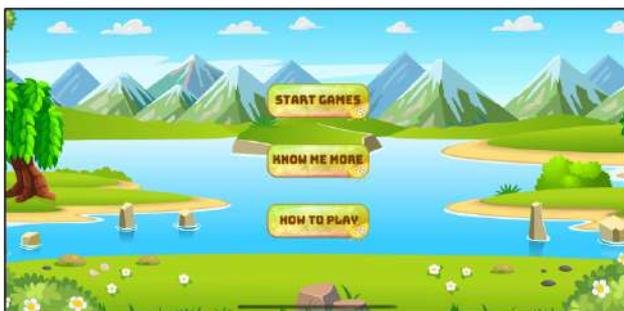
**Gambar 10. Sequence diagram**  
 Figure 10. Sequence diagram

### 7.3 Rancangan Antarmuka

Pada pengembangan *Game Edukasi Matching the Card* ini memiliki beberapa tampilan antarmuka yang terdapat didalam aplikasi *Game Edukasi* yang dibuat sesedehana mungkin dengan harapan pengguna awam hingga ahli dapat memahami penggunaannya, diantaranya adalah:

#### 1. Tampilan Menu Game

Pada gambar 11 menampilkan tampilan utama menu *game* saat melakukan *double* klik pada aplikasi, yang isinya terdiri dari Menu *Start Games*, *Know Me More*, dan *How To Play*.



**Gambar 11. Tampilan Menu Game**  
 Figure 11. Game Menu Display

#### 2. Tampilan Level

Pada gambar 12 Tampilan *Level* pada gambar menampilkan halaman saat memilih *menu Start Games* yang berisi tingkatan permainan yang terdiri dari *Easy*, *Medium*, dan *Hard*.



**Gambar 12. Level Display**  
 Figure 12. Level Display

#### 3. Tampilan Game

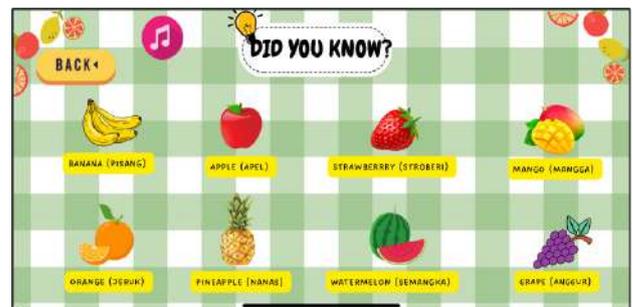
Pada gambar 13 tampilan *Game* pada menu *Pilih Level* menampilkan permainan yang akan di mainkan oleh pemain.



**Gambar 13. Tampilan Game**  
 Figure 13. Game Display

#### 4. Tampilan Menu Know Me More

Pada gambar 14 tampilan *game* menampilkan halaman objek gambar buah yang digunakan pada permainan



**Gambar 14. Tampilan Know Me More**  
 Figure 14. Know Me More Display

#### 5. Tampilan Menu How to play

Pada gambar 15 menampilkan cara bermain yang berisi keterangan atau aturan dari permainan yang akan dimainkan.



Gambar 15. Tampilan *How to Play*  
Figure 15. *How to Play Display*



Gambar 18. *Background* tampilan antarmuka pada start menu

Figure 18. *Background display interface on the start menu*

## 6. Tampilan *Did You Know*

Pada gambar 16 tampilan *Did You Know* menampilkan keterangan manfaat buah yang terdapat pada menu *Know Me More*.



Gambar 16. Tampilan *Did You Know*  
Figure 16. *Did You Know Display*



Gambar 19. *Background* tampilan antarmuka pada menu level

Figure 19. *Background display interface on menu level*

## 7.4 Material Collecting

Pada tahap ini akan dilakukan pengumpulan bahan pembuatan atau yang biasa yang di sebut *Asset* untuk Visualisasi *Game* Edukasi *Matching the Card*. Diantaranya:

### 1. *Button* (Tombol)

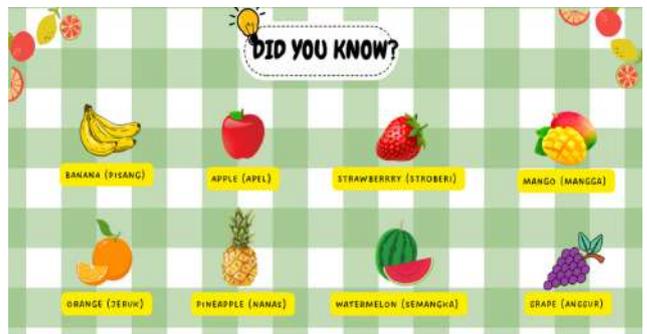
*Button* merupakan elemen antarmuka pengguna (*user interface*) yang digunakan untuk memicu suatu aksi atau perintah ketika diklik oleh pengguna. Seperti yang terlihat pada gambar 17.



Gambar 17. Desain *button* pada *game matching the card*  
Figure 17. *Button design in the matching the card game*

### 2. *Background* (Latar belakang)

*Background* adalah latar belakang tampilan antarmuka (UI) yang menjadi dasar atau latar visual dari elemen-elemen seperti tombol, teks, dan gambar. Seperti yang terlihat pada gambar 18, gambar 19, gambar 20, gambar 21 dan gambar 22.



Gambar 20. *Background* tampilan antarmuka pada menu *Did You Know*

Figure 20. *Background display interface on the Did You Know menu*



Gambar 21. *Background* tampilan antarmuka pada menu *Did You Know 2*

Figure 21. *Background display interface on the Did You Know 2 menu*

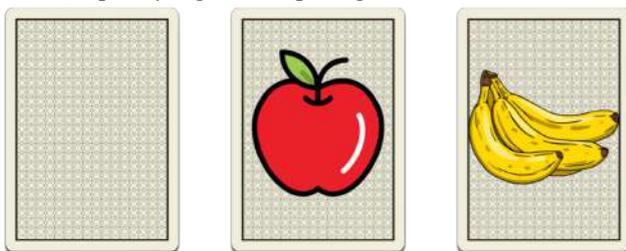


**Gambar 22. Background tampilan antarmuka pada How to Play**

*Figure 22. Background of the interface display in How to Play*

### 3. Deck kartu

Deck kartu pada aplikasi merujuk pada kumpulan kartu (seperti pada permainan kartu) yang ditampilkan, digunakan, atau dimanipulasi secara digital. Dalam konteks aplikasi, terutama game atau kuis, “deck” biasanya digunakan untuk game edukasi *matching the card*. “card deck” juga dipakai sebagai istilah desain, seperti tampilan kartu. Seperti yang terlihat pada gambar 23.



**Gambar 23. Tampilan antarmuka pada deck kartu**

*Figure 23. Interface view on card deck*

### 4. Score dan Timer

Asset score dan time adalah elemen penting dalam banyak aplikasi, terutama dalam game, yang digunakan untuk mengukur kinerja atau kemajuan pengguna. Seperti yang terlihat pada gambar 24.



**Gambar 24. Tampilan antarmuka pada Score dan Timer**

*Figure 24. Interface view on Score and Timer*

### 7.5 Assembly

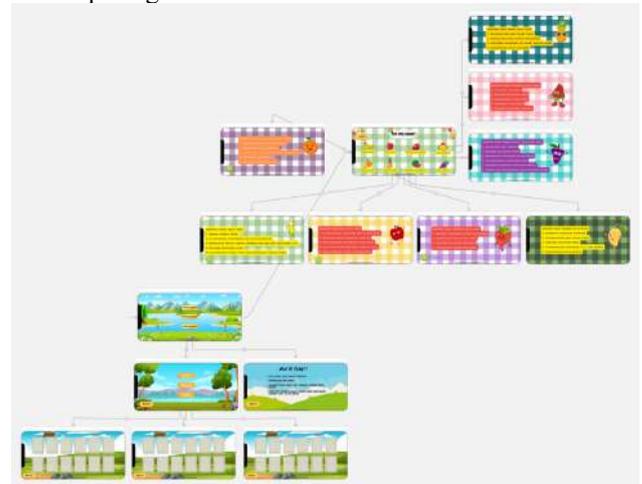
Pembuatan atau *Assembly* adalah tahapan Pengkodean dengan Implementasi Algoritma *Shuffle Random* dan Integrasi UI/UX dengan Mekanisme Permainan.

Pada tahapan pengkodean dengan Implementasi Algoritma *Shuffle Random*. Game ini melibatkan 12 tombol (kartu), *timer*, *skor*, dan *high score*. Dimana permainan berakhir saat semua kartu cocok atau waktu habis.

Penggabungan bahan – bahan yang telah dikumpulkan pada tahap sebelumnya, dan dibuat berdasarkan rancangan yang telah disusun pada tahap *design*.

Pada tahap ini semua bahan - bahan yang telah dikumpulkan sebelumnya akan dirangkai menggunakan *storyboard* pada *Xcode 16.3* untuk kemudian dirangkai

menjadi sebuah aplikasi game edukasi *Matching the Card*. Berikut adalah tampilan *storyboard* dari *Xcode 16.3* yang menunjukkan alur dan struktur antarmuka pengguna (UI) game edukasi *matching the card* berbasis iOS. Seperti yang terlihat pada gambar 26.



**Gambar 25. Storyboard pada Xcode 16.3**

*Figure 25. Storyboard in Xcode 16.3*

### 7.6 Testing

#### 7.6.1 Blackbox testing

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, perancangan dan pengkodean, salah satu metode Pengujian yang digunakan untuk menguji sistem Game Edukasi ini adalah metode pengujian *blackbox*.

Pada tahap *blackbox testing* berfokus pada persyaratan fungsional perangkat lunak seperti tombol atau sistem fungsional lainnya secara menyeluruh seperti: tombol mulai hingga hingga tombol hidup mati suara. pengujian *blackbox* terlihat pada tabel 5.

**Tabel 5. Blackbox Testing**

*Table 5. Blackbox Testing*

No.	Testing	Output Generated	Status		
			D1	D2	D3
1.	Click start game	Displays game levels	Succeed	Succeed	Succeed
2.	Click musik	Mute music	Succeed	Succeed	Succeed
3.	Click sound	Stop spelling	Succeed	Succeed	Succeed
4.	Click back	Displays exiting the game	Succeed	Succeed	Succeed
5.	Click how to play	Shows how to play the game	Succeed	Succeed	Succeed
6.	Click know me more	Displaying fruit objects in the game	Succeed	Succeed	Succeed
7.	Click permainan	Displays the game page	Succeed	Succeed	Succeed

### 7.6.2 White Box Testing

Pengujian selanjutnya adalah *white-box testing*. Pengujian difokuskan pada proses pengacakan menggunakan Algoritma *Shuffle Random* dengan potongan kode program seperti ditampilkan pada Gambar 27.

#### 1. Pengujian fungsi *OpenKartu()*

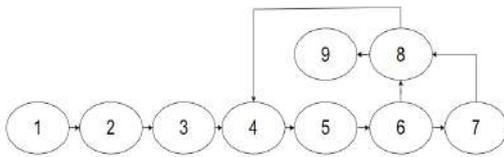
```

297 func openKartu() {
298     listKartu.shuffle()
299
300     let buttons: [UIButton] = [a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12]
301
302     for (index, button) in buttons.enumerated() {
303         let simbol = listKartu[index]
304         if let namaGambar = petaGambar[simbol] {
305             button.setImage(UIImage(named: namaGambar), for: .normal)
306         }
307     }
308 }
    
```

Gambar 26. kode program fungsi *openKartu()*

Figure 26. *openCard()* function program code

- 1 = 319
- 2 = 320
- 3 = 322
- 4 = 324
- 5 = 325
- 6 = 326
- 7 = 327
- 8 = 329
- 9 = 330



Gambar 27. *flow graph* fungsi *OpenKartu()*

Figure 27. *flow graph* of the *OpenKartu()* function

Berdasarkan hasil dari *flow graph* pada gambar 27, diketahui terdapat 9 *node* (n) dan 10 *edge* (€). Sehingga nilai CC adalah  $9 - 10 + 2 = 3$ . Berikut adalah jalur independen yang dihasilkan:

1. Jalur 1 : 1-2-3-4-5-6(true)-7-8-4-5-6(true)-7-8-9  
Keterangan : Semua gambar ditemukan di *petaGambar*, semua tombol menampilkan gambar.
2. Jalur 2 : 1-2-3-4-5-6(true)-7-8-4-5-6(false)-7-8-9  
Keterangan : Sebagian simbol ditemukan, sebagian tidak ditemukan (hasil campuran)
3. Jalur 3 : 1-2-3-4-5-6(false)-7-8-4-5-6(false)-7-8-9  
Keterangan : Tidak ada simbol ditemukan (semua tombol kosong)

Table 6. fungsi *OpenKartu()*

Table 6. *OpenKartu()* function

Kondisi	Hasil yang diharapkan	Keterangan
Semua simbol di ditemukan di <i>petaGambar</i>	Semua tombol (a1–a12) menampilkan gambar	Fungsi bekerja sempurna. Semua gambar tampil, <i>game</i> siap dimainkan.
Sebagian simbol ditemukan, sebagian tidak	Sebagian tombol menampilkan gambar, sebagian tidak menampilkan gambar (kosong)	Fungsi tetap berjalan. <i>PetaGambar</i> tidak lengkap.
Semua simbol tidak ditemukan di <i>petaGambar</i>	Semua tombol kosong (tidak menampilkan gambar sama sekali)	Fungsi berjalan, tapi tidak ada gambar. <i>Game</i> tidak dapat dimainkan.

### 7.6.3 Pengujian beta (beta testing)

Setelah melakukan pengumpulan data dari responden, kemudian data tersebut dihitung. Dalam cara menggunakan *System Usability Scale* (SUS) ada beberapa aturan dalam perhitungan skor SUS. Berikut ini aturan-aturan saat perhitungan skor pada kuesionernya:

1. Setiap pertanyaan bernomor ganjil, skor setiap pertanyaan yang didapat dari skor pengguna akan dikurangi 1.
2. Setiap pertanyaan bernomor genap, skor akhir didapat dari nilai 5 dikurangi skor pertanyaan yang didapat dari pengguna.
3. Skor SUS didapat dari hasil penjumlahan skor setiap pertanyaan yang kemudian dikali 2,5.

Aturan perhitungan skor untuk berlaku pada 1 responden. Untuk perhitungan selanjutnya, skor SUS dari masing-masing responden dicari skor rata-ratanya dengan menjumlahkan semua skor dan dibagi dengan jumlah responden. Berikut rumus menghitung skor SUS yang ditampilkan pada persamaan 1.

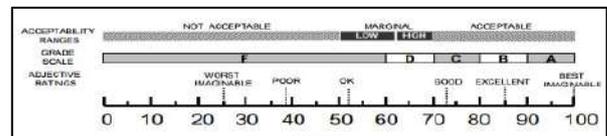
Setelah didapatkan isian kuesioner dari responden, kuesioner selanjutnya dihitung dengan menggunakan rumus perhitungan dari Q1 sampai Q10. Kemudian Jika sudah dapat jumlahnya, jumlah sebelumnya dikali dengan 2,5 untuk mendapatkan nilai akhir atau rata-rata. Berikut hasil dari perhitungan kuesioner dari data sebelumnya, dapat dilihat pada tabel 7.

Tabel 7. Perhitungan Skor SUS

Table 7. *SUS Score Calculation*

Skor Hasil Hitung										Jumlah	Nilai (Jumlah x 2.5)
Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10		
4	3	3	3	4	3	3	4	4	4	35	88
4	4	4	4	4	3	4	3	3	4	37	93
4	4	4	3	4	3	4	4	4	3	37	93
4	3	3	3	3	3	3	4	3	4	33	83
3	3	3	3	3	3	3	3	3	3	30	75
3	3	3	3	4	4	3	3	3	3	32	80
4	3	3	3	3	3	3	3	3	4	32	80
3	3	3	3	3	3	3	3	3	3	30	75
3	3	3	3	4	3	4	4	4	4	35	88
4	4	3	3	3	3	3	4	4	4	35	88
jumlah nilai (Jumlah x 2.5)											840
rata-rata = (nilai jumlah) / jumlah responden											84

Dari 10 responden didapat dengan jumlah rata-rata SUS yaitu 84. Hasil tersebut membuktikan bahwa pengujian pengembangan *game* edukasi *iOS matching the card* menggunakan algoritma *shuffle random* masuk ke dalam kategori *EXCELLENT* dengan *grade scale* A. Artinya secara *usability* berdasarkan data tersebut mendapatkan penilaian dapat diterima atau layak. *SUS Score* dapat kita lihat pada gambar 28.



Gambar 28. *SUS Score*

Figure 28. *SUS Score*

### 8. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan maka dapat diambil kesimpulan sebagai berikut:

1. Penelitian *game* edukasi *Matching the Card* dapat diselesaikan dengan menggunakan Metode *Multimedia Development Life Cycle*.
2. Telah melakukan penelitian di Sekolah Dasar Negeri 016 Tenggarong, Kelurahan Loa Tebu, Kecamatan Tenggarong, Kabupaten Kutai Kartanegara, Kalimantan

Timur dengan jumlah responden 10 siswa/I dengan skor penilaian *beta testing* nilai sebesar 84.

3. Pengujian *blackbox* yang dilakukan menunjukkan setiap tombol antarmuka pada aplikasi ini berjalan dengan baik dan pengujian *beta* menunjukkan tingkat kepuasan responden.
4. *Game* edukasi *Matching the Card* berbasis *iOS* dapat dimainkan dengan sistem operasi *iOS* versi 18.4.1.

## 9. Saran

Hasil penilaian *Game* edukasi *Matching the Card* masih jauh dari kata sempurna, terdapat beberapa elemen yang dapat di kembangkan lagi diantaranya:

1. Diharapkan *game* edukasi ini dapat memuat banyak pengenalan buah-buahan sehingga menambah ilmu pengetahuan tentang buah lebih luas.
2. Diharapkan untuk lebih dapat menyenangkan pemakaian.

## 10. Referensi

- ANANDA MUHAMAD TRI UTAMA. (2022). *RANCANG BANGUN SISTEM INFORMASI DESA BARU RAMBANG BERBASIS WEB*. 9, 356–363.
- Andrea, R. (2015). Teknik Pengacakan Posisi Objek Permainan Match-Up “Find Me! – Bumi Etam.” *Seminar Nasional Teknologi Informasi Dan Multimedia*, 3(1), 1–4.
- Bustomi, T., & Adytia, P. (n.d.). *Design And Development Of A Prototype For Information On The Availability Of Diesel Fuel At PT . Pinggan Wahana Pratama Through An IOT-Based Website Using NodeMCU Rancang Bangun Prototipe Informasi Ketersediaan Bahan Bakar Minyak Solar Pada PT . Pinggan Wahana Pratama Melalui Website Berbasis IOT NodeMCU*. 1–6.
- Desyanti, & Wetri Febrina. (2020). Pemodelan Unified Modelling Language (UML) dalam Pembuatan Aplikasi Data Penduduk. *SATIN - Sains Dan Teknologi Informasi*, 6(2), 56–66. <https://doi.org/10.33372/stn.v6i2.668>
- Firdausi, N. I. (2020). PENGEMBANGAN GAME EDUKASI MENGGUNAKAN SOFTWARE ADOBE FLASH CS6 PADA MATERI PENGENALAN MUFRADAT DI SEKITAR SEKOLAH DI PONPES MUSTHAFAWIYAH PURBABARU. *Kaos GL Dergisi*, 8(75), 147–154.
- Frasha, M. M. Y., & Masitoh, A. H. (2023). Aplikasi Manajemen Keuangan Pribadi pada Platform IOS Menggunakan Bahasa Pemrograman Swift. *Jurnal Esensi Infokom : Jurnal Esensi Sistem Informasi Dan Sistem Komputer*, 7(1), 29–35. <https://doi.org/10.55886/infokom.v7i1.574>
- Heri, F., Yunus, A., & Budianto, A. E. (2021). Penerapan Metode Algoritma Shuffle Random Pada Game 2D Pertualangan Pemuda Desa. *Kurawal - Jurnal Teknologi, Informasi Dan Industri*, 4(2), 155–168. <https://doi.org/10.33479/kurawal.v4i2.464>
- Kristianingrum, V., & Al-Fadillah, M. F. Y. (2022). Perancangan Website E-Commerce Penjualan Ikan Cupang. *JBMI (Jurnal Bisnis, Manajemen, Dan Informatika)*, 18(3), 164–180. <https://doi.org/10.26487/jbmi.v18i3.19538>
- Mustika, M., Sugara, E. P. A., & Pratiwi, M. (2018). Pengembangan Media Pembelajaran Interaktif dengan Menggunakan Metode Multimedia Development Life Cycle. *Jurnal Online Informatika*, 2(2), 121. <https://doi.org/10.15575/join.v2i2.139>
- Raharjana, I. K., & Justitia, A. (2015). Engineering Aplikasi Basis Data Pada Smartphone. *Jurnal Ilmiah Teknologi Informasi*, 13, 133–142.
- Rohmah, R. N. (2002). Case Tool Pengembangan Perangkat Lunak Berorientasi-objek menggunakan Unified Modeling Language ( UML ). *Jurnal Teknik Elektro Emitter*, 2(February), 39–42.
- Sukma, A. P., Yusuf, R., & Dai, R. H. (2023). Analisis Pengukuran Usability Sistem Informasi Manajemen Baznas ( Simba ) Menggunakan Metode System Usability Scale ( Sus ). *DIFFUSION Journal Of System And Information Technology*, 3(2), 224–231.
- Uminingsih, Nur Ichsanudin, M., Yusuf, M., & Suraya, S. (2022). Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula. *STORAGE: Jurnal Ilmiah Teknik Dan Ilmu Komputer*, 1(2), 1–8. <https://doi.org/10.55123/storage.v1i2.270>